Database Design Document: Student Course Enrollment System

1. Introduction

The Student Course Enrollment System is a comprehensive platform designed to modernize course administration for educational institutions. Key features include: a user-friendly student registration portal; instructor tools for managing enrollment and student grades; administrative modules for maintaining course and faculty data, ensuring data consistency; and student access to grade reports. The system streamlines student management, provides secure course information storage, and offers role-based access control.

2. Database Purpose

The primary goal of this database is to deliver a centralized, efficient, and reliable solution for managing all aspects of student course enrollment. Its key functions include:

- **Centralized Data Management:**

  Consolidates student records, course details, instructor information, enrollment data, schedules, and grades into a single, integrated system for seamless management.

- **Streamlined Enrollment Process:**

  Simplifies course registration by offering students an intuitive interface to search for, select, and enroll in courses with ease.

- **Efficient Course Management:**

  Equips instructors with tools to manage their courses effectively, including tracking student enrollment, assigning grades, and facilitating communication with students.

- **Administrative Oversight:**

  Empowers administrators to monitor and manage the entire enrollment process, oversee courses and instructors, generate reports, and maintain data accuracy and consistency.

- **Reporting and Analytics:**

  Provides actionable insights into key metrics such as enrollment trends, course popularity, and student performance to support data-driven decision-making.

3. Business Problems Addressed

This database addresses the following business problems related to student course enrollment:

- **Student Information Management:**

  Maintain accurate and comprehensive records of student details, such as personal information, declared majors, and contact information.

- **Course Offerings Management:**

  Organize and track course details, including course names, credit hours, capacity limits, semesters offered, departments, and assigned instructors.

- **Course Enrollment Facilitation:**

  Enable students to seamlessly enroll in courses, track enrollment dates, and manage enrollment records efficiently.

- **Prerequisite Management:**

  Define and enforce course prerequisites to ensure students meet the required knowledge criteria before enrolling in advanced courses.

- **Course Scheduling:**

  Provide a system for scheduling course sessions with details such as day of the week, start time, end time, and location.

- **Grade Recording:**

  Allow instructors to assign and store student grades securely within the system.

- **Waitlist Management:**

  Handle full-course scenarios by managing waitlists and automatically enrolling students when spots become available.

- **Major Management:**

  Track and manage academic majors along with their associated departments.

- **Instructor Management:**

  Store essential instructor information to facilitate course assignments and scheduling.

4. Business Rules

The database enforces the following business rules:

- **Student Information:**
    - Each student must have a unique StudentID.
    - Students must belong to a major.
- **Course Information:**
    - Each course must have a unique CourseID.
    - Courses must be associated with a department and a semester.
    - Each course must have a maximum capacity.
- **Enrollment:**
    - Students can only enroll in courses for which they meet the prerequisites.
    - Enrollment is limited by the course's capacity.
    - Enrollment records must be associated with a student and a course.
- **Scheduling:**
    - A course can have multiple schedule slots.
    - Schedule slots must have a day of the week, start time, end time, and location.
- **Grading:**
    - Each enrollment record may have one grade.
    - Grades must correspond to a valid grade letter and grade point.
- **Waitlist:**
    - Students can join a waitlist for a full course.
    - Students on the waitlist are enrolled in the order of their request date.
- **Department:**
    - Each course and instructor belong to one department.

5. Decision Designs

Key Design Decisions

- **Inclusion of the** Student **entity:** The Student entity is essential for storing student-specific information. It is the central element for login, enrolling in courses, and tracking academic progress.
- **Use of Associative Entities:** The many-to-many relationships between Course and CoursePrerequisite, and between Course and Schedule are resolved using the associative entities CoursePrerequisite and CourseSchedule. This ensures that relationships between courses and their prerequisites and courses and their schedules can be accurately represented without data redundancy.
- **Inclusion of the** Waitlist **entity:** The Waitlist entity is included to manage student requests to enroll in full courses. It helps to ensure that students are enrolled in courses in a fair and orderly manner when seats become available.
- **Foreign Keys:** The use of foreign keys ensures referential integrity across related entities. For example, the MajorID in the Student table links each student to their major in the Major table.
- **Third Normal Form (3NF):** All entities are designed to meet the requirements of the Third Normal Form (3NF) to minimize data redundancy and improve data integrity.

The following table provides a detailed overview of key design decisions:

Entity	**Why Included**	**Relationship to Other Entities**	
Student	Stores student information necessary for login, enrollment, and academic tracking.	Relates to ***Major*** (Many-to-One), ***Enrollment*** (One-to-Many), and ***Waitlist***	StudentID(PK), MajorID(FK), FirstName, LastName, Email, Password

Entity	Why Included	Relationship to Other Entities	
		(One-to-Many).	
Major	Stores information about academic majors and their affiliated departments. Ensures consistency in student major data.	Relates to ***Department*** (Many-to-One) and ***Student*** (One-to-Many).	MajorID(PK), DepartmentID(FK), MajorName
Instructor	Stores information about instructors, facilitating course scheduling and management.	Relates to ***Department*** (Many-to-One) and ***Course*** (One-to-Many).	InstructorID(PK), DepartmentID(FK), FirstName, LastName, Email, Password
Course	Stores course information, including name, credits, capacity, semester, and department. This is central to the	Relates to ***Semester*** (Many-to-One), ***Department*** (Many-to-One), ***Instructor*** (Many-to-One),	CourseID(PK), SemesterID(FK), DepartmentID(FK), InstructorID(FK), CourseName, Credits, Capacity

Entity	Why Included	Relationship to Other Entities	
	enrollment process.	***Enrollment*** (One-to-Many), ***CoursePrerequisite*** (Many-to-Many), and ***CourseSchedule*** (Many-to-Many), and ***Waitlist*** (One-to-Many).	
Enrollment	Records student enrollment in courses and tracks enrollment dates and grades.	Relates to ***Student*** (Many-to-One, Optional), ***Course*** (Many-to-One, Optional), and ***Grade*** (Many-to-One).	EnrollmentID(PK), StudentID(FK), CourseID(FK), GradeID(FK), EnrollmentDate
Semester	Stores information about academic semesters, defining the	Relates to ***Course*** (One-to-Many).	SemesterID(PK), SemesterName, StartDate, EndDate

Entity	Why Included	Relationship to Other Entities	
	time period during which courses are offered.		
Department	Stores basic information about departments, which is used to manage the affiliation of instructors, courses, and majors.	Relates to ***Course*** (One-to-Many), ***Instructor*** (One-to-Many), and ***Major*** (One-to-Many).	DepartmentID(PK), DepartmentName
CoursePrerequisite	Resolves the many-to-many relationship between courses and their prerequisites, ensuring that students have the required knowledge before enrolling in advanced courses.	Relates to ***Course*** (Many-to-Many - associative entity).	CourseID(PK,FK), PrerequisiteCourseID(PK,FK)
Schedule	Stores information	Relates to	ScheduleID(PK),

Entity	Why Included	Relationship to Other Entities	
	about class session time slots, including day of the week, start time, end time, and location.	***CourseSchedule*** (Many-to-Many - associative entity).	DayOfWeek, StartTime, EndTime, Location
CourseSchedule	Resolves the many-to-many relationship between courses and their schedules, linking courses to their scheduled class times.	Relates to ***Course*** (Many-to-Many - associative entity) and ***Schedule*** (Many-to-Many - associative entity).	CourseID(PK,FK), ScheduleID(PK,FK)
Grade	Stores detailed grade information, including grade letter and grade point, for calculating student GPA.	Relates to ***Enrollment*** (One-to-Many).	GradeID(PK), GradeLetter, GradePoint
WaitList	Records student waitlist applications for courses, enabling the	Relates to ***Student*** (Many-to-One) and	WaitListID(PK), StudentID(FK), CourseID(FK),

Entity	Why Included	Relationship to Other Entities	
	system to manage enrollment requests when courses are full.	***Course*** (Many-to-One).	RequestDate, Priority